

REMARKS

By this Amendment, Applicants add new claims 21 and 22, and hence claims 1-22 are all the claims pending in the application.

Applicants thank the Examiner for withdrawing the finality of the Office Action dated March 4, 2008. Since the finality of the Office Action dated March 4, 2008 was withdrawn, and since the arguments presented in Response dated May 5, 2008 with respect to the rejections under 35 U.S.C. § 103 were not considered, those arguments remain applicable. Therefore, Applicants will repeat and supplement those arguments where appropriate.

Claim Rejections - 35 U.S.C. § 103

Claims 1-3, 9-11, and 15 are rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over U.S. Patent No. 6,032,153 (Sadiq et al.) in view of MySQL 5.0 Reference Manual (hereinafter “MySQL”). Applicants respectfully traverse the rejection.

Sadiq is directed to a method of maintaining persistence in a shared object system. In Sadiq, a request is received to update a persistent record of a shard object's state. In response to the request, an SQL statement is dynamically generated using a data structure comprising data reflecting a portion of the state of the shared object and a map associating attributes of the shared object with the location of those attributes in a database. *See* Sadiq, Abstract. As a preliminary matter, the Examiner asserts that the request at column 2, lines 2-6 allegedly teaches the claimed “receiving a request that affects an item.” Accordingly, the Examiner asserts that the “shared object” of Sadiq teaches the claimed “item.”

In the Office Action, the Examiner also asserts that Sadiq allegedly teaches the claimed feature of “identifying a respective type of the item.” The Examiner’s position is based on column 2, lines 2-6 of Sadiq and the “value type” at column 4, lines 13 to 15 of Sadiq. As discussed above, Sadiq discloses that an SQL statement is dynamically generated using a map associating attributes of the shared object with the location of those attributes in a database. *See also*, Sadiq, col. 3, line 66 - col. 4, line 2. Sadiq describes that each record in the map comprises an attribute name, a column name, a value type, a table name, a key type, and a system assigned key indicator. *See* Sadiq, col. 4, lines 6-8. Specifically, Sadiq states that the “value type refers to the data type of the attribute.” *See* Sadiq, col. 4, lines 14-15.

Accordingly, Sadiq neither teaches nor suggests “identifying a respective type of the item,” as recited in claim 1. Rather, at best, Sadiq discloses identifying a type of an attribute of an item. Indeed, Sadiq merely discloses that the shared objects and their associated attributes are maintained in a distributed object system. *See* Sadiq, col. 3, lines 18-28. However, Sadiq is silent as to the classification of the objects themselves, and discloses nothing about the objects, which have associated attributes, as having an associated “type.”

As a result, Sadiq fails to teach the claimed feature of “identifying a respective type of the item.” MySQL similarly fails to teach such a feature and merely discloses changing the type of existing columns, which would at best contain attributes values of an item. *See* MySQL, p. 1. Therefore, the combination of Sadiq and MySQL fails to teach “identifying a respective type of the item.”

Further, the Examiner asserts that Sadiq allegedly teaches the claimed feature of “retrieving a set of attributes based on the type of the item and a partial structured query language statement corresponding to the attributes.” The Examiner’s assertion depends on the position that the traversing of a data structure (56) containing attributes of an item at column 6, lines 31 to 35 of Sadiq teaches “retrieving a set of attributes,” as recited in claim 1. *See* Office Action, page 4. Sadiq describes that data structure (56) contains an object identifier and all other attributes that have been changed as a result of a transaction. *See* Sadiq, col. 5, lines 34-38, col. 6, lines 13-16. Further, data structure (56) is used along with map (60), discussed above, to generate SQL statements used to update a database. *See* Sadiq, col. 6, lines 20-23; FIGS. 3B, 3D. To generate the SQL statement, persistence service (48) traverses data structure (56) and generates a single SQL statement for all attributes in the data structure (56). *See* Sadiq, col. 6, lines 32-35.

However, Sadiq neither teaches nor suggests “retrieving a set of attributes based on the type of the item.” Rather, Sadiq describes object 52 has the attributes of part number, price, in stock, order number relationship, and order item identity. *See* Sadiq, col. 6, lines 3-5; FIG. 3A. For each attribute, a dirty indicator is used to indicate whether the attribute has been changed by a transaction, and only those attributes that have been changed are updated in the database. *See* Sadiq, col. 4, lines 54-62. Sadiq further describes that data structure (56) is generated using all attributes that have changed, as indicated by the dirty identifier, and an object identifier. *See* Sadiq, col. 6, lines 12-17.

As a result, Sadiq neither teaches nor suggests “retrieving a set of attributes based on the type of the item,” but rather the set of attributes retrieved to create data structure (56) in Sadiq are, at best, based on an object identifier and attributes that are marked as dirty by the dirty indicator. *See* Sadiq, col. 5, lines 40-45. Sadiq describes that the object identifier is merely an identifier of the object itself. *See* Sadiq, col. 6, lines 7-8. Similarly, the retrieved attributes themselves cannot be the type of the item since they merely represent current values of the changed attributes as indicated by the dirty indicator. Therefore, the attributes of Sadiq are not retrieved based on a type of the item.

Moreover, the Examiner asserts that the value type, which is a data type of an attribute such as integer, real, Boolean, string, or character, allegedly teaches the claimed “type of the item.” Even if the Examiner’s assertion is true, Applicants respectfully submit that Sadiq neither teaches nor suggests “retrieving a set of attributes based on the type of the item.” As discussed above, the basis for retrieval of attributes in Sadiq is not whether the attribute itself is an integer, real, Boolean, string, or character, but rather the basis for retrieval of attributes of an item is merely, at best, an object identifier and attributes that are marked as dirty by the dirty indicator.

Accordingly, for at least the above reasons, Sadiq fails to teach “retrieving a set of attributes based on the type of the item.”

MySQL also fails to teach such a feature, and hence the combination of Sadiq and MySQL fails to teach the claimed feature of “retrieving a set of attributes based on the type of the item and a partial structured query language statement corresponding to the attributes.”

Therefore, for at least the above reasons, the combination of Sadiq and MySQL fails to teach or suggest all the features of claim 1, and hence the combination of Sadiq and MySQL would not have rendered claim 1 and its dependent claims unpatentable.

Claims 9, 10, and 11 recite features similar to those discussed above, and hence the combination of Sadiq and MySQL would not have rendered claims 9, 10, and 11, or any of their dependent claims, unpatentable for at least analogous reasons.

Claims 4 and 20 are rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Sadiq in view of MySQL, and further in view of Seaman et al. (U.S. Patent App. 2003/0093433, hereinafter "Seaman"). Applicants respectfully traverse the rejection.

Claims 4 and 20 depend on claims 1 and 11, respectively, and incorporate all the features of claims 1 and 11. Seaman is cited for teaching retrieving a portion of an insert statement. Even if Sadiq and MySQL could have somehow been modified based on Seaman, as the Examiner asserts in the Office Action, the combination would still not contain all the features of claims 1 and 11, and hence claims 4 and 20, as discussed above. Accordingly, the combination of Sadiq, MySQL, and Seaman would not have rendered claims 4 and 20 unpatentable.

Claims 5 and 19 are rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Sadiq in view of MySQL, and in further view of Wildermuth (U.S. Patent 5,950,188). Applicants respectfully traverse the rejection.

Claims 5 and 19 depend on claims 1 and 11, respectively, and incorporate all the features of claims 1 and 11. Wildermuth is cited for teaching retrieving information that indicates access

rights for the structured query language statement. Even if Sadiq and MySQL could have somehow been modified based on Wildermuth, as the Examiner asserts in the Office Action, the combination would still not contain all the features in claims 1 and 11, and hence claims 5 and 19, as discussed above. Accordingly, the combination of Sadiq, MySQL, and Wildermuth would not have rendered claims 5 and 19 unpatentable.

Claim 6 and 12-14 are rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Sadiq in view of MySQL, and further in view of Reiner (U.S. Patent 6,219,676). Applicants respectfully traverse the rejection.

Claims 6 and 12-14 depend on claims 1 and 11, respectively, and incorporate all the features of claims 1 and 11. Reiner is cited for teaching the elements of claims 6 and 12-14. Even if Sadiq and MySQL could have somehow been modified based on Reiner, as the Examiner asserts in the Office Action, the combination would still not contain all the features of claims 1 and 11, and hence claims 6 and 12-14, as discussed above. Accordingly, the combination of Sadiq, MySQL, and Reiner would not have rendered claims 6 and 12-14 unpatentable.

Claims 7, 8, and 16-18 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Sadiq in view of MySQL, and further in view of Reiner et al. (U.S. Patent 6,219,676). Applicants respectfully traverse the rejection.

Claims 7 and 8, and 16-18 depend on claims 1 and 11, respectively, and incorporate all the features of claims 1 and 11. Reiner et. al is cited for teaching cursors. Even if Sadiq and MySQL could have somehow been modified based on Reiner et. al, as the Examiner asserts in the Office Action, the combination would still not contain all the features in claims 1 and 11, and

hence claims 7, 8, and 16-18, as discussed above. Accordingly, the combination of Sadiq, MySQL, and Reiner et. al would not have rendered claims 7, 8, and 16-18 unpatentable.

New Claims

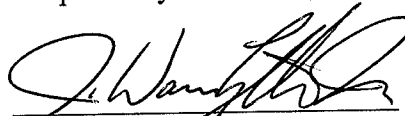
As discussed above, Applicants add new claims 21 and 22. Applicants respectfully submit that claims 21 and 22 should be deemed patentable at least by virtue of their dependency. Applicants also respectfully submit that the references cited by the Examiner fail to teach or suggest all the features of claims 21 and 22.

Conclusion

In view of the above, reconsideration and allowance of this application are now believed to be in order, and such actions are hereby solicited. If any points remain in issue which the Examiner feels may be best resolved through a personal or telephone interview, the Examiner is kindly requested to contact the undersigned at the telephone number listed below.

The USPTO is directed and authorized to charge all required fees, except for the Issue Fee and the Publication Fee, to Deposit Account No. 19-4880. Please also credit any overpayments to said Deposit Account.

Respectfully submitted,



J. Warren Lytle, Jr.
Registration No. 39,283

SUGHRUE MION, PLLC
Telephone: (202) 293-7060
Facsimile: (202) 293-7860

WASHINGTON DC SUGHRUE/142133

46159

CUSTOMER NUMBER

Date: July 8, 2008